345 Ludic Computing

Tutorial 5

Behaviour Trees & Adaptation

Simon Colton & Alison Pease
Computational Creativity Group
Department of Computing
Imperial College London

ccg.doc.ic.ac.uk

Question 1

Design behaviour trees for the following NPCs.

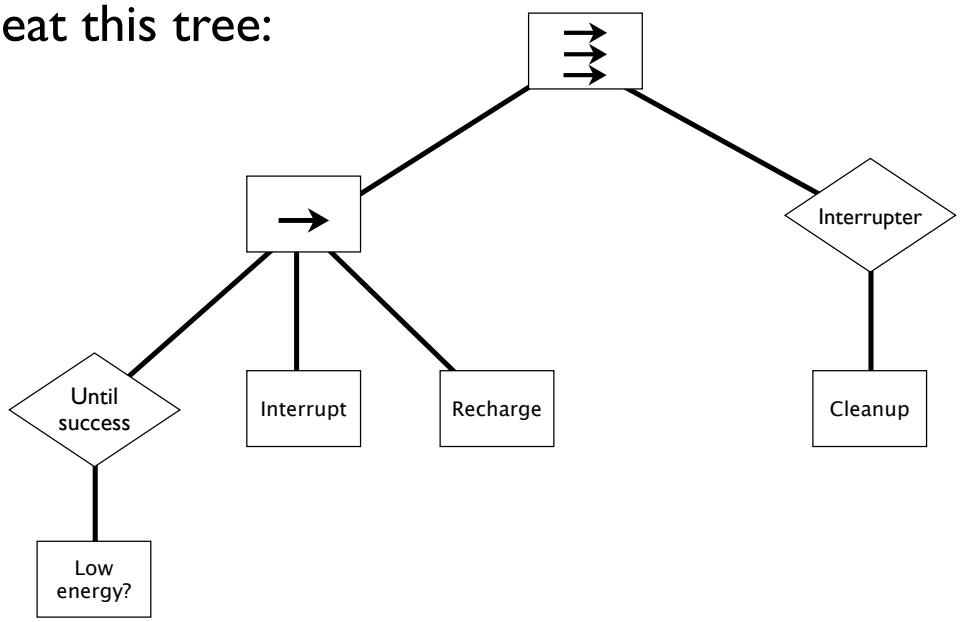
a) Two guards patrol a number of rooms together, sometimes waiting in a room before moving on to the next. If they spot the player then they both attack, although guard 2 first locates and sets off the wall-mounted security alarm in that room.

b) The cleaner robot from the lecture, as specified by the hierarchical state machine. Give two sub-trees for the recharge and cleanup behaviours, co-ordinated by a third main tree.

Solution 1b

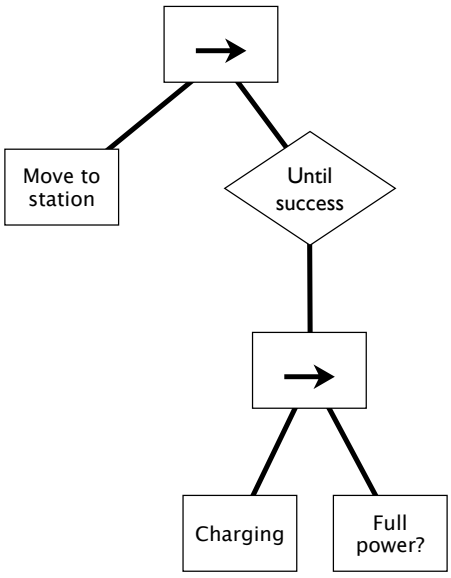
Main Tree

Repeat this tree:

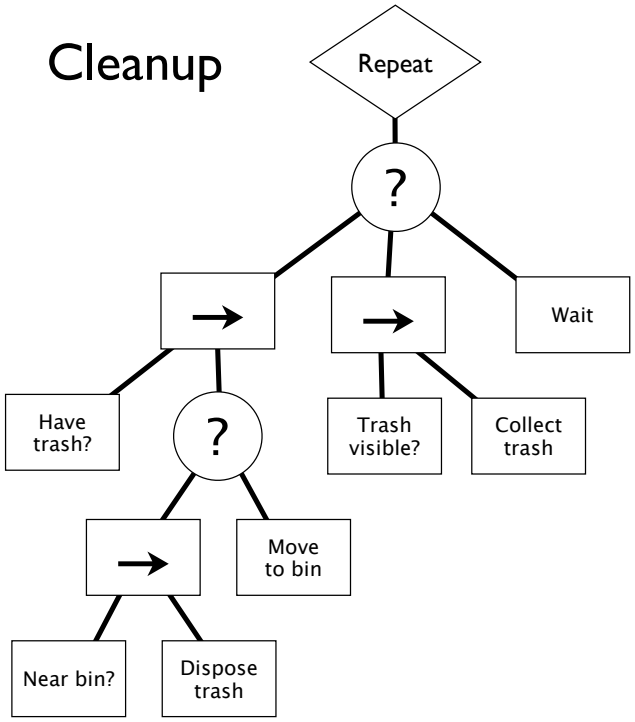


Solution 1b

Recharge



Cleanup



Question 2

a) A player can move L(eft) and R(ight) in game of Space Invaders. Construct a hierarchical 4-gram table based on the input LLLRL RLRLL LLRLR LRRRL LLRRR LRLRR.

b) Predict the next action for all possible inputs of length 3, using a sample threshold of 5. Give the n-value used and probability for each prediction.

Solution 2

2 x 1-Grams
Sample 30

L	16
R	14

4 x 2-Grams
Sample 29

LL	7
LR	9
RL	8
RR	5

8 x 3-Grams
Sample 28

LLL	4
LLR	3
LRL	6
LRR	3
RLL	2
RLR	6
RRL	2
RRR	2

16 x 4-Grams
Sample 27

LLLL	1	RLLL	2
LLLR	3	RLLR	0
LLRL	2	RLRL	4
LLRR	1	RLRR	2
LRLL	1	RRLL	1
LRLR	5	RRLR	1
LRRL	0	RRRL	2
LRRR	2	RRRR	0

Predictions
Threshold 5

	Next	n	Prob.
LLL	L	3	0.57
LLR	L	3	0.67
LRL	R	4	0.83
LRR	L	2	0.61
RLL	L	3	0.57
RLR	L	4	0.67
RRL	R	3	0.75
RRR	L	2	0.61

Question 3

An RTS game agent bases it's strategy it's strength and territory relative to the player, using the following model of the game

States: SL (strong-large), SS (strong-small), WL (weak-large), WS (weak-small)

Transitions: A(ttack), E(xplore), R(etreat), B(uild) (The state graph is complete.)

The agent uses Q-learning to adapt to a new player. All Q-values start at 1, $\alpha = \gamma = 0.75$.

a) Describe the Q function after the experiences <WS, B, 0.5, SS>, <SS, E, 0.6, SL>, <SL, A, 1.0, SL>, <SL, R, -1.0, SS>

b) Why are these initial Q-values unsatisfactory?

c) Why might you want to change α as training progresse

Solution 3a

If <s, a, r, s'> then $Q(s, a) = (1 - \alpha)Q(s, a) + \alpha(r + \gamma \max(Q(s', a'))$

<WS, B, 0.5, SS> **Q(WS, B)** = 0.25*1 + 0.75(0.5 + 0.75*1) = 1.19

<SS, E, 0.6, SL> **Q(SS, E)** = 0.25*1 + 0.75(0.6 + 0.75*1) = 1.26

<SL, A, 1.0, SL> **Q(SL, A)** = 0.25*1 + 0.75(1 + 0.75*1) = 1.56

<SL, R, -1.0, SS> Note that max(Q(SS, ?)) = Q(SS, E) = 1.26 so we get **Q(SL, R)** = 0.25*1 + 0.75(-1 + 0.75*1.26) = 0.21

So after the first four experience tuples all Q-values are still 1, except for the four values given above.

Solution 3b/c

b) Initially all actions are valued equally in all states, which will make the agent behave randomly until it adapts. It might be better to start with Q-values that reflect a generic strategy before adapting it to a new player.

c) The learning rate α decides how much new experiences are favoured over old. A high α is useful initially to kick start learning. But if our player is unlikely to change then we might want to lower α as learning progresses. This will protect our accumulated experience, i.e. we will require a greater amount of evidence to change our strategy.